

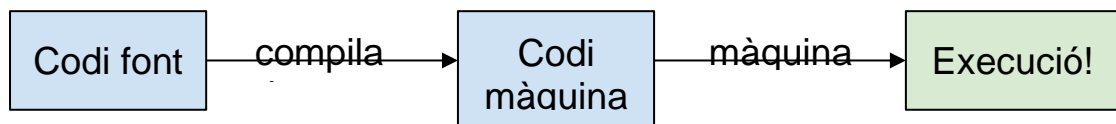
| | |
|-----------------------------------------------|---|
| Models d'execució de codi. | 1 |
| Mecanismes d'execució de codi en un navegador | 2 |
| Capacitats i limitacions d'execució | 2 |

Models d'execució de codi.

Si els llenguatges de programació es classifiquen quant a la manera d'executar-se, diferenciar dos grans tipus:

- **Compilats:** aquells que necessiten d'un compilador com a pas previ a l'execució, no fent immediat l'accés al codi font. Tot i això, els llenguatges compilats acostumen a ésser més ràpids que els interpretats en temps d'execució.

Exemples: C, Java (a Bytecode), Pascal...

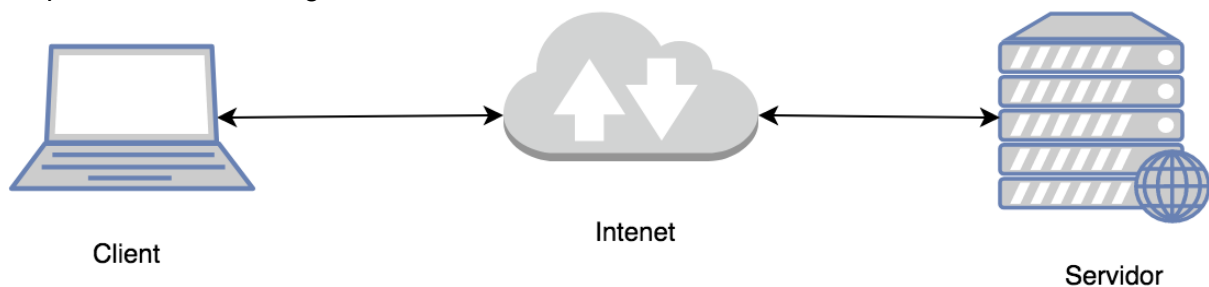


- **Interpretats:** aquells que s'executen mitjançant un intèrpret, que processa les comandes que inclou el programa una a una, fent que siguin menys eficient en temps d'execució que els compilats. Tot i això, tenen l'avantatge que el codi font és públic, ja que no cal compilar-lo per a la seva execució.

Exemples: Javascript, Python, PHP, Ruby...



Tanmateix, donada una arquitectura client-servidor, els llenguatges de programació també es poden classificar segons on s'executen:



- **Client:** Principalment gestionen l'aparença de la pàgina web i el seu contingut, però poden interactuar amb el servidor i dades locals, tals com les galetes.

Exemples: HTML, JavaScript, CSS, Ajax, jQuery...

- **Servidor:** Acostumen a executar tasques relacionades amb el processament de peticions de clients, enviament de pàgines web, estructuració d'aplicacions web, integració amb servidors i bases de dades, gestió de les bases de dades, codificació de dades a HTML...

Exemples: Java, Python, PHP, Ruby...

Mecanismes d'execució de codi en un navegador

Tot i que cada navegador disposa de la seva pròpia arquitectura, la gran majoria disposen de blocs comuns anomenada arquitectura de referència. Aquesta arquitectura està formada pels següents blocs:

- **Interfície d'usuari:** cadascuna de les parts visibles del navegador, menys la que mostra la pàgina web.
- **Motor de cerca:** carrega una adreça determinada i suporta els mecanismes bàsics com anar una pàgina endarrere, endavant o tornar a carregar la mateixa pàgina. També gestiona les alertes JavaScript que es mostren a l'usuari i el procés de càrrega d'una pàgina nova.
- **Motor de renderització:** s'encarrega de generar la pàgina web que s'ha demanat al servidor a partir del codi que li ha arribat a través d'internet, establint les dimensions exactes a mostrar de cada mòdul.
- **Intèrpret JavaScript:** és motor encarregat d'interpretar el codi JavaScript al navegador, seguint l'estàndard ECMAScript.
- **Persistència de dades:** funciona com magatzem de dades al navegador per emmagatzemar galetes, certificats, historial i sessions d'usuari entre d'altres.

Capacitats i limitacions d'execució

JavaScript, presenta certes avantatges i limitacions d'execució. Tot seguit es llisten les principals:

Avantatges:

- Requereix de menys interacció amb el servidor

Limitacions:

- JavaScript pot ser desactivat per l'usuari

- Reacciona amb la interacció amb l'usuari
- Modifica estils i continguts
- Pre-processa dades al client
- Multiplataforma, pot ser executat per gairebé qualsevol combinació de sistema operatiu i navegador web
- No interacciona directament amb els llenguatges del servidor
- No pot accedir a: les preferències del navegador, finestra...
- No pots accedir al sistema de fitxers del client
- Cada navegador web disposa d'un intèrpret diferent, que pot produir resultats lleugerament diferents segons el navegador.